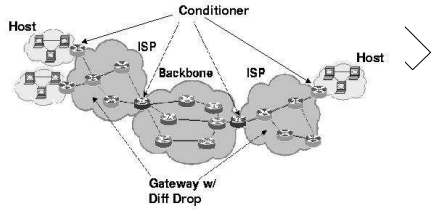
On Detecting SLA Violations in QoS (and non-QoS) Network Domains: A Measurement-based Approach

With A. Habib, S. Avasarala, V. Prabhakar and Prof. B. Bhargava
E-mail: fahmy@cs.purdue.edu
http://www.cs.purdue.edu/homes/fahmy/
http://www.cs.purdue.edu/homes/fahmy/cerias/

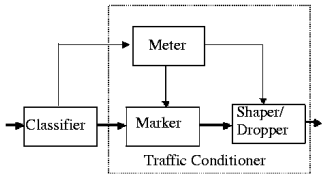
The End-to-End Argument

- Keep end-to-end operations at end points
- Minimal processing at core network routers
- What about un-trusted end points?
  - Use edge-to-edge monitoring

Differentiated Services (DiffServ)

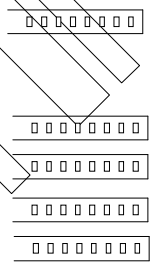


Traffic Conditioner [RFC 2475]



Per Hop Behaviors at Core Routers

- Expedited Forwarding
  - Like a virtual leased line
  - zero loss
- Assured Forwarding
  - 4 classes of packets
  - 3 drop precedences per class



(Distributed) Denial of Service

Users are no longer trusted ⇒ 13,000 denial of service attacks recorded in 3 weeks by CAIDA

- IP trace-back
- Filtering spoofed packets
- Throttles

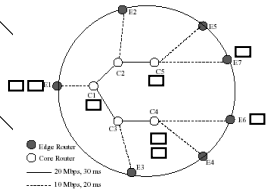
- On-line monitoring of network activity is required to maintain confidence in the security and QoS of networks

Goals

- Monitor network for unusual patterns
- Measure delay, link loss, and throughput achieved by user inside a network domain
- Excessive traffic changes internal characteristics inside a domain
- Detect attacks *or need for re-configuration* on-line based on performance degradation (losses, delays)
- As with any mechanism, attackers may disable monitoring mechanisms, but they need to gain control of many nodes

Challenges

- How to measure all parameters without involvement of internal routers, i.e. Edge-to-edge
- Low overhead monitoring that does not alter the traffic being monitored



QoS Parameter Measurements

1. Delay Measurements

- Copy user packet headers with a probability *p*, and probe the network
- $Delay^i = t^i_{egress} - t^i_{ingress}$
- Calculate average delay using Exponential Weighted Moving Average
- Needs clock synchronization between ingress and egress

QoS Parameter Measurements

2. Throughput Measurements

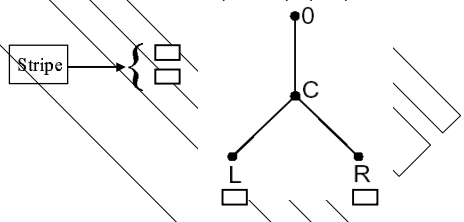
- Egress routers have the current aggregate rate per user (useful for DDoS)
- Egress can be polled, if necessary, to get bandwidth achieved by a user

QoS Parameter Measurements

3. Loss Measurements

- Core routers watch packet loss. Recording drop at core increases overhead on the core
- How to obtain *loss ratio* without core routers?
- Adapt end-to-end network tomography techniques

Striped Unicast Probing



Stripe: Fleet of packets sent with no inter-packet transmission delay.

All packets of a stripe should have similar experience inside network.

Definitions

- $\alpha_k(D)$: Probability that all packets in D are successfully transmitted to node k given that they arrived at k's parent.
- $\beta_k(D|D')$: Conditional probability that packets in D are successfully transmitted to node k, given that those in D′ are successfully transmitted to k (again given they made it to the parent).
- $\beta_k(D|D') = \alpha_k(D \cup D') / \alpha_k(D')$
- With perfect correlations among D and D′ β_k would be 1.

Inference with Binary Stripes

Send <L, R> stripe from root to leaves

$$Z_D = \begin{cases} 1 & \text{if all packets in D reach their destination} \\ 0 & \text{otherwise} \end{cases}$$

$$EZ_{12} = \alpha_C(12)\,\alpha_L(1)\,\alpha_R(2)$$
$$= \alpha_C(2)\,\beta_C(1|2)\,\alpha_L(1)\,\alpha_R(2)$$

Similar equations for EZ_1 and EZ_2

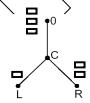
Taking $\beta_c = 1$, $\alpha_C = Z_1Z_2/Z_{12}$, $\alpha_L = Z_{12}/Z_2$, $\alpha_R = Z_{12}/Z_1$

Is $\beta_c = 1$? No.

Since $\beta_c \leq 1 \Rightarrow$ we overestimate α_C and underestimate α_L and α_R

Enhancing Stripe Correlations

- Send three-packet stripe <L, R, R>
- More likely that packet 1 reaches C upon reception of packets 2 and 3 at receivers R rather than reception of packet 2 alone.



$$\frac{EZ_{123}}{EZ_{23}} = \alpha_L(1) \times \beta_C(1|23)$$

If $\beta_C(D|D') \geq \beta_C(D|D'')$ *for all* $D'' \subset D'$

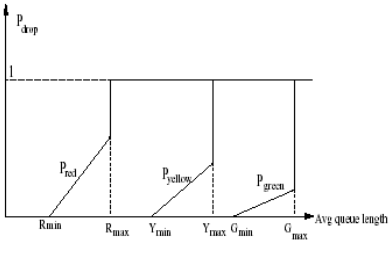
Then wider stripe reduces estimate of α_C, countering bias of α_L, α_R

α_R is estimated similarly with complementary stripe <R, L, L>

Loss Inference for QoS Networks

- QoS-enabled networks have multiple classes of traffic
- Queues can have multiple drop precedences
- Can we extend loss inference using probes with multiple drop precedences?

RED Queue with 3 Drop Precedences



Overall Loss Inference

$$L_{overall} = \frac{\Sigma N_i P'_i L_i}{\Sigma N_i}$$

- N_i = Number of stripes sent of drop precedence i
- P'_i = Probability of drop at the queue. Can be calculated from queue parameters
- L_i = 1- A_k (probability of transmission)
- i = 1 to number of drop precedences

SLA Violation Detection

- $Delay^i > SLA^i_{delay}$?
- $Loss^i > SLA^i_{loss}$?
- $BW^j > SLA^j_{BW}$?

For each type of traffic, derive statistical bound for delay and loss

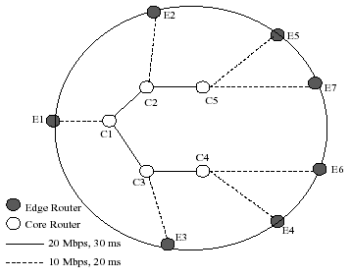
Communication Overhead Reduction

- Edge-to-edge monitoring scheme needs to probe the network continuously to detect attacks in real time
- Can we probe on demand to reduce communication overhead?
- How frequently we should probe for delay and loss?

On Demand Probing

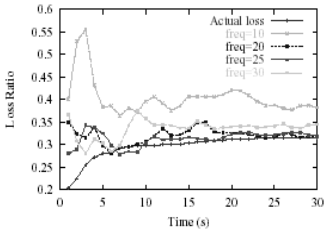
- Monitor the network for delay with probability *p*
- If $delay^i > SLA^i_{delay}$ Monitor the network for loss
- *Which egress pair to probe for loss*? Delay probe is one. Leaf nodes with Least Common Ancestor of the spanning tree topology
- If $loss^i > SLA^i_{loss}$ Monitor the network for throughput
- If $BW^j > SLA^j_{BW}$ Flow *j* is violating SLA by taking excess resources. Attacks by injecting packets from several entry points can be detected by SLA_{BW} violation detection
- Detect links with high loss. Check IP *destination* address of exiting flows for possible DDoS attack at the downstream domain

Simulation Topology

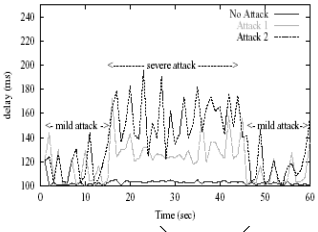


Multiple hosts are connected to each edge router.

Loss Inference

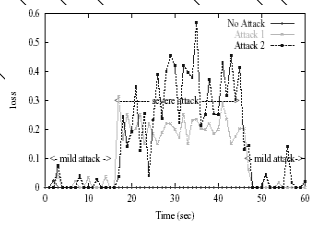


Delay Variation during Attacks



Attack1 causes 25-30% loss, Attack causes loss more than 35%

Loss Inference during Attacks



Loss variation follows the pattern of delay variation

Conclusions

- Use simple probability to infer excessive loss at certain points in the network. 3-packet stripe shows good correlation. 2 shows less correlation and 4 does not improve much. Use for different drop precedences
- Overhead can be reduced by probing on demand when delays/losses increase
- Monitor network and detect service violations and attacks OR need for re-configuration